Streaming Optimised Scientific Software: an Introduction to EESSI

ISC'24 tutorial proposal

Sebastian Achilles[1], Kenneth Hoste[2] and Alan Ó Cais[3]

[1]Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany
[2]HPC-UGent, Ghent University, Belgium
[3]Centre Européen de Calcul Atomique et Moléculaire (CECAM), Switzerland

Abstract

What if there was a way to avoid having to install a broad range of scientific software from scratch on every HPC cluster or cloud instance you use or maintain, without compromising on performance?

Installing scientific software for supercomputers is known to be a tedious and time-consuming task. Especially as the HPC user community becomes more diverse, computational science expands rapidly, the diversity of system architectures increases the application software stack continues to deepen. Simultaneously, we see a surge in interest in cloud computing for scientific computing. Delivering optimised software installations and providing access to these installations in a reliable, user-friendly, and reproducible way is a highly non-trivial task that affects application developers, HPC user support teams, and the users themselves.

This tutorial aims to address these challenges by providing the attendees with the knowledge to stream optimised scientific software. For this, the tutorial introduces European Environment for Scientific Software Installations (*EESSI*), a collaboration between various European HPC sites & industry partners, with the common goal of creating a shared repository of scientific software installations that can be used on a variety of systems, regardless of which flavor/version of Linux distribution or processor architecture is used, or whether it's a full size HPC cluster, a cloud environment or a personal workstation.

We cover the usage of EESSI, different ways to accessing EESSI, how to add software to EESSI, and highlight some more advanced features. We will also show how to engage with the community and contribute to the project.

Topic area: Programming Environments & System Software, Introductory Tutorial

Keywords: Computing Infrastructure, Containers in HPC, Reproducibility, Scientific Software Development, System Software & Runtime Systems

1 Detailed Description of the Tutorial

Overview and Goals

Application developers, HPC sites, and end users spend significant amounts of time on optimised software installations. Surveys conducted at the *"Getting Scientific Software Installed"* Birds-of-a-Feather sessions that we (co-)organised at both SC and ISC reveal that this (still) involves a significant amount of 'manual' effort. In the SC'19 survey, less than half of the respondents consistently automate software installation, and only 25% automate environment module file generation. Despite these ubiquitous problems, there is still inadequate collaboration between HPC sites to date: less than 30% of respondents indicated that they work together with other HPC sites regarding software installation, even in most recent surveys. Hence, an EESSI tutorial is very relevant to ISC'24 attendees as this tool helps relieve these burdens and fosters collaboration.

The **European Environment for Scientific Software Installations (EESSI)**[1] project is a collaborative project between different partners in the HPC community supported by the MultiXscale EuroHPC Centre of Excellence to build a common stack of optimised scientific software installations for everything from laptops to big HPC systems and cloud infrastructures. The project uses EasyBuild to install software on top of a compatibility layer, and uses CernVM-FS to distribute the software installations to client systems.

EasyBuild is used by well over 100 HPC sites worldwide (incl. JSC, CSCS, the Digital Research Alliance of Canada, LUMI, ...), has over 125 unique contributors on a yearly basis, and an active Slack channel with over 830 members. It has reached a critical mass with a welcoming and active community guaranteeing continued development and support, and a growing user community.

EESSI is an initiative built on the foundations of the EasyBuild community, and goes one step further by providing a truly uniform software stack. It already has a wide spectrum of collaborators despite being a relatively young project.

EESSI is motivated by the observation that the landscape of computational science is changing in various ways. Additional families of general-purpose microprocessors including Arm 64-bit (aarch64) and RISC-V on top of th well-established Intel and AMD processors (both x86_64), and different types of GPUS (NVIDIA, AMD, Intel) are increasing the diversity in system architectures. The rapid expansion of computational science beyond traditional domains like physics and computational chemistry, including bioinformatis, Machine Learning (ML) and Artificial Intelligence (AI), etc. leads to a significant growth of the software stack that is used for running scientific workloads. The emergence of commercial cloud infrastructure (Amazon EC2, Microsoft Azure, ...) and private cloud infrastructure (OpenStack) has competitive advantages over on-premise infrastructure for computational workloads, such as near-instant availability, increased flexibility, a broader variety of hardware platforms, and faster access to new generations of microprocessors. In addition the manpower that is available in the HPC user support teams that are responsible for helping scientists with running the software they require on high-end (and complex) infrastructure like supercomputers (and beyond) is limited. These reasons indicate that there is a strong need for more collaboration on building and installing scientific software to avoid duplicate work across computational scientists and HPC user support teams.

The main goal of EESSI is to provide a collection of scientific software installations that work across a wide range of different platforms, including HPC clusters, cloud infrastructure, and personal workstations and laptops, without making compromises on the performance of that software.

Software installations included in EESSI are optimized for specific generations of microprocessors by targeting a variety of instruction set architectures (ISAs), like for example `x86_64` processors

[1] https://eessi.io

supporting the AVX-512 instructions, and Arm processors that support SVE instructions.

To provide optimized installations of scientific software stacks for a diverse set of system architectures, the EESSI project consists of 3 layers, which are constructed by leveraging various open source software projects:

- The filesystem layer uses CernVM-FS to distribute the EESSI software stack to client systems
- The compatibility layer levels the ground across different (versions of) the Linux operating system (OS) of client systems that use the software installations provided by EESSI. It consists of a limited set of libraries and tools that are installed in a non-standard filesystem location (a "prefix"), which were built from source for the supported CPU families using Gentoo Prefix.
- The top layer of EESSI is called the software layer, which contains the actual scientific software applications and their dependencies. Building, managing, and optimising the software installations included in the software layer is done using EasyBuild, a well-established software build and installation framework for managing (scientific) software stacks on High-Performance Computing (HPC) systems. Next to installing the software itself, EasyBuild also automatically generates environment module files. These files, which are essentially small Lua scripts, are consumed via Lmod, a modern implementation of the concept of environment modules which provides a user-friendly interface to end users of EESSI. The initialisation script that is included in EESSI automatically detects the CPU family and microarchitecture of a client system by leveraging either `archspec`, a small Python library, or `archdetect`, a minimal pure bash implementation of the same concept.

The tutorial covers several critical aspects of EESSI, including its motivation, goals, and high-level design, as demonstrated through hands-on demonstrations and guided illustrations. The main features and functionalities of EESSI are highlighted, as well as the different ways to access the software stack, such as native installation, use in an HPC production setup, or without administrator privileges. The tutorial also covers various use cases for EESSI, including portable workflows, integration into a CI environment, adding software to the EESSI, and also configuring GPU support. Extensive instructor-led examples are used to put the theoretical knowledge into practice.

The organisers have successfully worked together on a number of papers, presentation materials, and hands-on tutorials on the topic under consideration several times in recent years. A common template will guarantee a coherent visual appearance, and using our previous experience, we will provide high-quality content and materials. Moreover, the presenters have experience in organising tutorials in the context of SC and ISC, which helps to understand expectations and how to structure the material to help ISC attendees to get the most out of it.

Target audience

This tutorial is intended for

- End users who want to empower themselves to use a uniform software stack without compromising on performance, on top of what is provided centrally by the HPC support team;
- System managers, administrators and user support teams, responsible for the operational aspects of HPC systems and concerned about hardware optimised scientific software installations;
- Cloud and CI users, who want to use a common stack of optimised scientific software installations;
- System manufacturers and integrators interested in state-of-the-art software installation tools, who want to leverage the collective expertise incorporated in EasyBuild and EESSI.

Outline

We believe that a half-day format would be most appropriate for this tutorial, as it would allow ample time for guided examples and hands-on practice, in addition to providing a comprehensive overview of EESSI. The tutorial outline for the half-day format is included below.

Half-day format, 3.5 hours of tutorial content

09:00-09:30 *(30min)* Introduction to EESSI: Motivation + goals (10min), High-level design + inspiration (15min), EuroHPC MultiXscale CoE (funding) (5min)

09:30-10:15 *(45min)* Using EESSI, incl. hands-on: Using a native installation (10min), Hands-on (20min), Alternative access mechanisms: containers, cvmfsexec (15min)

10:15-10:45 *(30min)* Use cases (workflow, CI), incl. hands-on

10:45-11:00 *(15min)* Testing software in EESSI

11:00-11:30 *(30min coffee break)*

11:30-12:10 *(40min)* Adding software to EESSI, incl. hands-on

12:10-12:40 *(30min)* GPU support, incl. hands-on

12:40-12:50 *(10min)* EESSI community

12:50-13:00 *(10min)* Q&A + closing remarks

Hands-on and presentation materials

A significant portion of the tutorial time is used for guided examples and hands-on, as they are fundamental to exposing the benefits of EESSI. This highlights the practical nature of the tutorial. Before each example session the theoretical concepts are explained through a consistent set of presentation materials. The guided examples will be carried out in a pre-configured environment on virtual machines in the Cloud that can be accessed by the attendees from their laptops using SSH. The examples cover accessing EESSI in different ways such as native, in a production HPC setup as well as inside containers, the use cases of EESSI, adding support for new software to EESSI, troubleshooting, GPU support, selected advanced features, and how to contribute to EESSI. The guided examples will happen mostly in the proximity of a coffee break allowing the attendees to continue a bit during the coffee break should they want to and ensuring the maximum benefit from the tutorial.

The tutorial is based on previous tutorials archived at https://github.com/EESSI/docs/tree/main/talks/20231205-Introduction-to-EESSI-CASTIEL2, but will be updated with new developments in EESSI and EasyBuild. This tutorial will also be made available on that site, and will also refer to the EESSI documentation on https://www.eessi.io/docs. It should be noted that even though this tutorial has never been presented at ISC, all the authors have extensive experience in other tutorials and workshops, both inside and outside the ISC and SC conference series. Additionally, they engaged in other collaborative efforts within the EESSI and EasyBuild community. Providing coherent and streamlined tutorial is well within the authors' experience.

2 Logistics

Content level

60% beginner, 40% intermediate

Prerequisites

Attendees wishing to participate in the guided examples are expected to use their own notebook computers with a working SSH client.

Detailed information on how to prepare for this tutorial will be provided through the tutorial website, similar to https://tutorial.easybuild.io/2022-isc22.

Expected attendance

Based on attendance at previous EESSI and EasyBuild events, including the yearly EasyBuild User Meetings (over 30 attendees in the last physical edition in 2023, close to 100 in 2021 and 2022 in an online setting) and previous related tutorials (PRACE-VI-SEEM 2017 Spring School with 25-30 attendees, a free online EasyBuild tutorial in June'20 with about 100 attendees, the half-day ISC'21 tutorial with over 25 attendees, and the in-person (half-day) ISC'22 EasyBuild tutorial with about 20 attendees, and an online EESSI tutorial in Dec'23 had over 130 registrations), we anticipate this EESSI tutorial to be well attended with 20-50 attendees.

Travel support

The tutorial presenters request free passes for their tutorial, as well as travel support for two presenters, one from Spain, one from Belgium.

3 CVs

3.1 CV for Sebastian Achilles

Jülich Supercomputing Centre (JSC)
Forschungszentrum Jülich GmbH
Wilhelm-Johnen-Straße
52425 Jülich (Germany)
s.achilles@fz-juelich.de

Education

M.Sc. Simulation Sciences, RWTH Aachen University.

B.Sc. Physics, 2014, RWTH Aachen University.

Professional Experience

2022 - present: Lead of Software Team, Jülich Supercomputing Center (JSC), Forschungszentrum –present Jülich GmbH, Jülich.

2022 - present: HPC and Cloud System engineer, Jülich Supercomputing Center (JSC), Forschungszentrum –present Jülich GmbH, Jülich.

2020 - 2022: Research Assistant, Jülich Supercomputing Center (JSC), Forschungszentrum –present Jülich GmbH, Jülich.

2017 - 2020: Postgraduate student, Aachen Institute for Advanced Study in Computational Engineering Science (AICES), RWTH Aachen University, Aachen.

Tools

EasyBuild: (scientific) software build and installation framework

Release Manager (2022 - present)

Developer (2021 - present)

https://easybuild.io

Conferences, Workshops and Tutorials

Maintaining a Modern Scientific Software Stack Made Easy with EasyBuild tutorial at ISC'22, June 2022 (https://easybuild.io/tutorial/isc22)

Selected Publications

Conference Papers

Estela Suarez, Wolfgang Frings, Norbert Attig, Sebastian Achilles, Jacopo De Amicis, Thomas Eickermann, Eric Gregory, Björn Hagemeier, Andreas Herten, Jenia Jitsev, Dorian Krause, Edoardo Di Napoli, Jan Meinke, Kristel Michielsen, Bernd Mohr, Dirk Pleiter, Alexandre Strube, and Thomas Lippert. *Developing Exascale Computing at JSC.* NIC Symposium

2020. Vol. 50. Publication Series of the John von Neumann Institute for Computing (NIC) NIC Series. NIC Symposium 2020, Jülich (Deutschland).

Journal Papers

Jonas B. Hauck, Carsten Honerkamp, Sebastian Achilles, and Dante M. Kennes. *Electronic instabilities in Penrose quasi-crystals: competition, coexistence and collaboration of order.* Aug. 31, 2020.

Xiao Zhang, Sebastian Achilles, Jan Winkelmann, Roland Haas, André Schleife, and Edoardo Di Napoli. *Solving the Bethe-Salpeter equation on massively parallel architectures.* June 15, 2020.

3.2 CV for Alan Ó Cais

Centre Européen de Calcul Atomique et Moléculaire (CECAM)
Avenue de Forel 2, BCH 3103
1015 Lausanne, Switzerland
alan.ocais@cecam.org

Education

Ph.D Theoretical Physics, 2005, Trinity College Dublin (Ireland)

M.Sc. High Performance Computing, 2002, Trinity College Dublin (Ireland)

B.Sc. Theoretical Physics, 2001, Trinity College Dublin (Ireland)

Professional Experience

2022 - present: Scientific Software Engineer with CECAM

2016 - 2021: E-CAM Centre of Excellence Software Manager at Jülich Supercomputing Centre (Germany)

2010 - 2016: Research Fellow at Jülich Supercomputing Centre (Germany)

2008 - 2010: Scientific Coordinator at The Cyprus Institute (Cyprus)

2006 - 2008: Research Fellow at University of Adelaide (Australia)

2005 - 2006: Research Fellow at Trinity College Dublin (Ireland)

Tools and Projects

MultiXscale EuroHPC Centre-of-Excellence
steering committee member, technical manager, and work package leader (2023–present)
https://www.multixscale.eu

European Environment for Scientific Software Installations (EESSI)
active contributor (2020–present)
https://eessi.io - https://eessi.io/docs

EasyBuild: (scientific) software build and installation framework
developer & co-maintainer (2013–present)
https://easybuild.io - https://docs.easybuild.io

Conferences, Workshops and Tutorials

Streaming Optimised Scientific Software: an Introduction to EESSI, online tutorial in context of MultiXscale EuroHPC CoE, co-organisor, December 2023
https://github.com/EESSI/docs/tree/main/talks/20231205-Introduction-to-EESSI-CASTIEL2

Best Practices for CernVM-FS in HPC, online tutorial in context of MultiXscale EuroHPC CoE, co-organisor, December 2023
https://multixscale.github.io/cvmfs-tutorial-hpc-best-practices

Maintaining a Modern Scientific Software Stack Made Easy with EasyBuild tutorial at ISC'21, June 2021 (https://easybuild.io/tutorial)

Organiser of the first *European HPC Training Stakeholder Workshop* (in coordination with the European Commission)

Instructor at all E-CAM Centre of Excellence *Extended Software Development Workshops*

Local organizer of the *2nd EasyBuild User Meeting (EUM'17)*. Jülich. Germany.

Invited instructor at PRACE seasonal training workshops:

PRACE Autumn School in HPC Programming Techniques, Athens 25-28 November 2014: Lectures on Profiling and Optimization and Benchmarking

PRACE Winter School, Tel Aviv 10-13 February 2014: Lectures on Profiling and Optimization and Benchmarking

Joint HP-SEE, LinkSCEEM-2 and PRACE HPC Summer Training, Athens 13-15 July 2011: Lectures on Profiling and Optimization and Benchmarking

Selected Publications

Conference Papers

A. Ó Cais, P. Steinbach, *Expanding user communities with HPC Carpentry.* Proceedings of the Workshop on HPC Education and Training for Emerging Technologies at ISC2019, Journal of Computational Science Education. 11. 21-25.

D. Alvarez, A. O'Cais, M. Geimer, K. Hoste, *Scientific Software Management in Real Life: Deployment of EasyBuild on a Large Scale System.* Proceedings of the 3rd International Workshop on HPC User Support Tools, Salt Lake City, USA, 2016.

Journal Papers

B. Dröge, V. Holanda Rusu, K. Hoste, C. van Leeuwen, A. O'Cais, T. Röblitz, *EESSI: A cross-platform ready-to-use optimised scientific software stack.* Software: Practice and Experience, 53(1):176-210, Jan. 2022, https://doi.org/10.1002/spe.3075

MJT Oliveira, N Papior, Y Pouillon, V Blum, E Artacho,. . . , *The CECAM electronic structure library and the modular software development paradigm.* J Chem Phys. 2020;153:024117

M.S. Mahbub, (Adelaide U. & Rajshahi U.), Alan O'Cais, Waseem Kamleh, B.G. Lasscock, Derek B. Leinweber, Anthony G. Williams, *Isolating Excited States of the Nucleon in Lattice QCD*, Phys.Rev.D80:054507,2009.

M.S. Mahbub, (Adelaide U. & Rajshahi U.) , Alan O'Cais, Waseem Kamleh, Ben G. Lasscock, Derek B. Leinweber, Anthony G. Williams, (Adelaide U.), *Isolating the Roper Resonance in Lattice QCD*, Phys.Lett.B679:418-422,2009.

Alexandrou, C., et al., *Evaluation of fermion loops applied to the calculation of the n' mass and the nucleon scalar and electromagnetic form factors*, Computer Physics Communications 183.6 (2012): 1215-1224.

J. Foley, K. Jimmy Juge, A. O'Cais, M. Peardon, S.M. Ryan and J. I. Skullerud, *Practical all-to-all propagators for lattice QCD*, Comput. Phys. Commun. 172 (2005) 145.

3.3 CV for Kenneth Hoste

HPC-UGent - Ghent University
Krijgslaan 281 - building S9, 9000 Ghent (Belgium)
kenneth.hoste@ugent.be

Education

PhD in Engineering: Computer Science, 2010, Ghent University (Belgium)

Masters in Computer Science, 2005, Ghent University (Belgium)

Professional Experience

2010–present: HPC System Administrator, Ghent University (Belgium)

Tools and Projects

MultiXscale EuroHPC Centre-of-Excellence
project partner and work package leader (2023–present)
https://www.multixscale.eu

European Environment for Scientific Software Installations (EESSI)
active contributor (2020–present)
https://eessi.io - https://eessi.io/docs

EasyBuild: (scientific) software build and installation framework
lead developer & release manager (2010–present)
https://easybuild.io - https://docs.easybuild.io

Conferences, Workshops and Tutorials

Streaming Optimised Scientific Software: an Introduction to EESSI, online tutorial in context of MultiXscale EuroHPC CoE, co-organisor, December 2023
https://github.com/EESSI/docs/tree/main/talks/20231205-Introduction-to-EESSI-CASTIEL2

Best Practices for CernVM-FS in HPC, online tutorial in context of MultiXscale EuroHPC CoE, co-organisor and presenter, December 2023
https://multixscale.github.io/cvmfs-tutorial-hpc-best-practices

Introduction to CernVM-FS, online tutorial, co-presenter, January 2021
https://cvmfs-contrib.github.io/cvmfs-tutorial-2021

Maintaining a Modern Scientific Software Stack Made Easy with EasyBuild tutorial at ISC'21 + ISC'22, co-organisor, https://tutorial.easybuild.io/2022-isc22

(co-)organisation of *EasyBuild User Meeting* workshops in Ghent (2016), Jülich (2017), Amsterdam (2018), Louvain-la-Neuve (2019), Barcelona (2020), online (2021 + 2022), and London (2023), https://easybuild.io/eum

co-organisaton of FOSDEM devroom on *HPC, Big Data & Data Science* (2014, 2016–2023), https://archive.fosdem.org/2023/schedule/track/hpc_big_data_and_data_science

Online EasyBuild tutorial, June 2020, https://tutorial.easybuild.io/2020-06-isc20/

(co-)organisation of *Getting Scientific Software Installed* Birds-of-a-Feather sessions at ISC'13, SC'13, ISC'14, SC'14, SC'15, SC'18, SC'19

Modern Scientific Software Management using EasyBuild & co, invited tutorial at PRACE-VI-SEEM 2017 Spring School, Nicosia (Cyprus), April 2017 (`https://events.prace-ri.eu/event/601` - presentation - recording part 1, part 2)

Introduction to EasyBuild: Getting Scientific Software Installed With Ease, invited tutorial at HPC Advisory Council 2016, Lugano (Switzerland), March 2016 (`https://www.hpcadvisorycouncil.com/events/2016/swiss-workshop` - presentation part 1, part 2 - recording part 1, part 2)

(co-)organisation of 11 EasyBuild hackathons across Europe + Austin (US) (2011–2016)

co-organisation of the 1st International Workshop on *HPC User Support Tools* (HUST-14) at SC'14, New Orleans (USA), Nov. 2014

Selected Publications

Conference Papers

P. Forai, K. Hoste, G. Peretti-Pezzi, B. Bode, *Making Scientific Software Installation Reproducible On Cray Systems Using EasyBuild*. Cray User Group (CUG) 2016, London, UK, 2016.

D. Alvarez, A. O'Cais, M. Geimer, K. Hoste, *Scientific Software Management in Real Life: Deployment of EasyBuild on a Large Scale System*. Proceedings of the 3rd International Workshop on HPC User Support Tools, Salt Lake City, USA, 2016.

M. Geimer, K. Hoste, R. McLay, *Modern Scientific Software Management Using EasyBuild and Lmod*. Proceedings of the 1st International Workshop on HPC User Support Tools, New Orleans, USA, 2014.

K. Hoste, J. Timmerman, A. Georges, S. De Weirdt, *EasyBuild: building software with ease*. High Performance Computing, Networking, Storage and Analysis (SC Companion 2012), Salt Lake City, USA, 2012.

K. Hoste, L. Eeckhout, *COLE: Compiler Optimization Level Exploration*. Proceedings of the 6th International Symposium on Code Generation and Optimization (CGO 2008), Boston, USA, 2008.

Journal Articles

B. Dröge, V. Holanda Rusu, K. Hoste, C. van Leeuwen, A. O'Cais, T. Röblitz, *EESSI: A cross-platform ready-to-use optimised scientific software stack*. Software: Practice and Experience, 53(1):176-210, Jan. 2023, `https://doi.org/10.1002/spe.3075`

K. Hoste, L. Eeckhout, *A Methodology for Analyzing Commercial Processor Performance Numbers*. IEEE Computer, 42(10):70-76, Oct. 2009.

K. Hoste, L. Eeckhout, *Microarchitecture-Independent Workload Characterization*. IEEE Micro, 27(3):63-72, May 2007.